Report No. 395

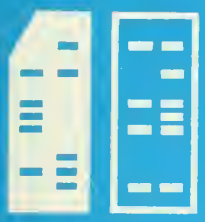
ILLIAC IV

QUARTERLY PROGRESS REPORT

January, February, and March 1970

Contract No.
USAF 30(602)-4144

ILLIAC IV Document No. 208



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

ILLIAC IV

QUARTERLY PROGRESS REPORT

January, February, and March 1970

Contract No.
USAF 30(602)-4144

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois
61801

April 15, 1970

This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.

TABLE OF CONTENTS

REPORT SUMMARY

The prototype processing element (PE) has been completely debugged and as a result five circuit card types have been re-designed to operate properly. In addition, three circuit card types have been re-designed in an attempt to increase the operating speed of the PE. Production diagnostics for printed circuit cards continue to be delivered to Burroughs from the University of Illinois.

Production PE's are being held up because of quality control problems with externally manufactured printed circuit cards.

The PEX Control Computer (PDP-9) was delivered and installed at Burroughs in February but has not been used due to the delays in the production PE's.

As a result of the January 7 meeting (composed of Software personnel and future ILLIAC IV users) held in Chicago to study programming languages for ILLIAC IV, it was decided to "shelve" TRANQUIL and commence a study of FORTRAN extensions for ILLIAC IV. GLYPNIR and ASK continue to be refined.

The problem of providing utility programs in needed areas is receiving considerable attention.

A simulation of the proposed Operating System II indicated that it would contribute to a more efficient use of system resources than Operating System I.

Specifications for graphics equipment were written and distributed to vendors. Data-communication hardware requirements were investigated.

The ILLIAC IV Users Group, FOURUM, will become the main vehicle for disseminating information to potential ILLIAC users.

Project expenditures through February 1970 are as follows:

Burroughs Corporation	$21.8 million
University of Illinois	$ 4.4 million

There are currently 127 people on the ILLIAC IV Staff. The construction of the Center for Advanced Computation Building is progressing according to schedule.

1. HARDWARE

1.1 Logic Simulation and Diagnostics

1.1.1 Logic Simulation

1.1.1.1 PE Simulator

The PE Simulator was used heavily this period for verification of functional tests (e.g., divide and multiply operations) and verification of the combinational tests.

Debugging of the PE was successfully completed, and the assistance of the PE Simulator contributed towards having error-free tests to run on the PE Exerciser.

A few minor errors, due to the logical design, were found and removed from the PE Simulator.

1.1.1.2 CU Simulator System

The CU/PE Card Simulator Programs were modified to facilitate the generation and compilation of the good and bad board simulators automatically. In addition, several programs of the system were subjected to various modifications to facilitate correct simulation of all the cards generated to date.

Work on the CU Section Simulator continued. If a simulator for a section of the Control Unit is generated by simply combining the appropriate card simulators, its inefficiency is apparent because the connections among the cards in a section are so complex that each card must be simulated many times. To avoid this inefficiency, it was decided to reduce the complexity of the connections by partitioning each card simulator into two or three parts. Since three partitioning algorithms were devised, the effectiveness of each was compared by applying it to

the TMU section [1]. According to the results, the simplest algorithm was adopted and the programming of the section simulator generator has been started.

Wherever necessary the card simulator generator programs have been rewritten to suit the requirements of the section simulator; this particularly applies to the programs which sort by signal names and package locations (WEDTR1 and WEDTR2). The WEDTR1 program accepts the backplane wiring and produces an additional error listing file which can be examined to correct for I/O connections. The WEDTR2 program takes into account the partitioning of the cards.

All the cards in the TMU section were partitioned (incorporating corrections to the netlist wherever necessary) and trial runs were made on the new section simulator generator.

1.1.2 Card Test Generation

1.1.2.1 Card Test Generation System

During the quarter this system was fully debugged and some features to enable more flexible use were added. It is presently being used on a production basis to generate diagnostics for PE and CU cards.

1.1.2.2 Card Test Translation System

The test translation system consists of two programs which translate failure and pattern input information into test dictionaries for each CU and PE card. The test translation programs, called CUFAIL/XREF and PEFAIL/XREF, produce a cross-reference listing for input and output patterns and failure modes. For each test, the initial flip-flop settings are listed, the applied test pattern is given, the number of clocks applied is indicated, and each output pin has associated with it the particular failures causing that output to differ from the correct value, that value being the output with no failures. A second listing is given of those tests which detect each failure mode. A third

listing is given of the equivalent failures on each card. The major difference between the two programs is that the formats of the CU card dictionary must be much more compact, due to a larger number of tests and failure modes.

These dictionaries represent the primary tool of finding failures on PE cards and CU cards, and in interpreting the results of the TI 561 System, which exercises the PE cards, and the PEX System, which will exercise the CU cards. Twelve dictionaries have successfully been run for CU cards and sixteen dictionaries for PE cards.

The system for automatically generating code for the TI 561 tester for PE cards has been improved and has been used on a production basis for sixteen PE card types. A program for automatic generation of PEXTAP code for testing CU cards is under development, but is not yet operational.

1.1.2.3 Generation of PE Card Tests

All but one of the thirty-eight PE board types have had diagnostics completed or are presently waiting for Engineering Change Notes (ECN's) from Burroughs. The diagnostics detect all possible single failures on these boards. Sixteen boards have had code for the TI 561 tester and test dictionaries generated and forwarded to Burroughs. Eight boards are waiting for ECN's and thirteen boards are waiting for TI 561 code generation.

Of the seven MLU boards, six have had diagnostics completed and are awaiting TI 561 code generation. One board requires completion of diagnostics.

1.1.2.4 Generation of CU Card Tests

Twelve CU boards have had diagnostics completed. These diagnostics detect between 98 per cent and 100 per cent of potential single failures on the board. On several of these boards the diagnostics indicate probable areas of redundant logic design by Burroughs. Test dictionaries have been run on these cards and are being forwarded to Burroughs.

1.1.3 PE Diagnostics

1.1.3.1 Path Test Generation

Detection and location phase path tests were released to Burroughs at the beginning of this period. A demonstration of the failure location capability by temporary insertion of failures was successful. Evaluation of the usefulness of these tests for the PE off-line diagnosis will be made during the PE production process.

1.1.3.2 Combinational Test Generation

Combinational test generation began in earnest this quarter. The Combinational Test Generator (CTG) and PEXTAP Input Generator (PIG) programs were modified slightly to reflect changes elsewhere in the system. Several tests associated with the barrel switch (BSW) were generated and delivered to Burroughs. First attempts at other tests were halted at the PEXTAP assembler due to the number of data patterns exceeding the length declared in assembler arrays, thus making it necessary to split tests into varying numbers of subtests.

At the present time, tests have been completed for the multiplier decoder gates (MDG) and carry propagate adder (CPA), in addition to the barrel switch. Some subtests of the address adder (ADA) are complete. The remaining ADA tests and multiplicand select gates (MSG) tests are near completion.

Source language and compiled versions of the circuit analysis programs, ECAP and CIRCUS, were received. These are to be used as an aid in circuit failure analysis. Results from a sample circuit used as a test on CIRCUS did not compare closely with the results in the manual. The source language file requires a special program to transfer properly from tape to disk.

1.1.3.3 Control Logic Test Generation

The test generation for the PE control logic, consisting of seven subsystem programs, has progressed in this quarter, emphasis being placed on the modification and completion of the last two programs.

The main functions of the equation transformation program, one of the two programs, is finding conflicts, value assignment and sneak path checking. It produces the test patterns for the control logic. There are 111 Boolean equations for the control logic. These 111 equations are transformed into about 1800 arc descriptions by the equation transformation program. These 1800 arc descriptions are divided up into four groups of arc descriptions and fed into the PGM program in four runs. The four outputs of PGM consist of 464 paths and are combined and rearranged. The rearranged PGM paths are fed into the test generation program to produce the test patterns for the control logic. The documentation of the algorithm of conflict finding and sneak path checking is under preparation.

1.1.3.4 Functional Test Generation

University of Illinois personnel verified Burroughs' functional tests using the PE simulator. In the intermediate stage of verification of the functional tests, simulation outputs were sent back to Burroughs; University of Illinois personnel also assisted in analyzing errors and evaluating the tests.

The University personnel received the final version of the functional tests from Burroughs in the middle of the quarter and had verified about seventy per cent of them by the end of the quarter.

Further verification and evaluation will be continued during the next period.

1.1.4 PEX Control Computer and Supervisor System

The PDP-9/L computer was installed in Paoli during February. After several wiring mistakes in the PEX were corrected, tests and results could be transmitted between the PEX and the PDP-9. Several minor bugs were discovered: (1) the University had assumed that the Clear Test Number signal would also clear the PE Error indicator; this was not true, and some minor PEX rewiring was necessary; (2) the PEX frequently acknowledged receipt of a signal when it had not actually begun to act upon it; the PDP-9 promptly removed the signal upon receiving the acknowledgment and the action was never carried out; (3) the PEX could not operate properly with the PDP-9 using the REPEAT/COMPARE/HALT instruction--after an error halt it was impossible to start the next test without clearing the PEX, but that would destroy the comparison data showing which bits had been in error before this data could be read into the PDP-9 and printed. A minor bug in the supervisory system software was also discovered and corrected.

This system was installed in PEX #1, since PEX #2 was being used to debug the prototype PE. Due to the delays in completion of the production PE's, the system has not yet been used.

2.1 Operating System

2.1.1 Operating System I

The Creeper module uses execution lists generated by the Job Parser to call system modules needed by the job. The Operator module handles SPO commands controlling job execution and performs system bookkeeping for each job. These two modules have been coded and simulation of a system with dummy slave modules has been accomplished to debug the controlling functions of these two central modules. The Disk Manager, which handles queueing of disk allocation requests, has been coded and debugged. The Data Processor, which effects transfer of files from the B6500 disk to the ILLIAC disk, has been coded, including preliminary data conversion routines, and debugging is in progress. The Collector and Loader, previously written and debugged, have been incorporated into the system. The functions of the Execution Monitor have been specified in close connection with the Job Partner and Hardware Sub-System (HSS). Coding of this last module has begun.

Coding of the Job Partner has begun now that the Hardware Sub-System intrinsics have been specified. Likewise, the macro assembler has made possible the preliminary specification of the OS4 I/O and utility requests; OS4 itself is nearing completion.

The HSS provides the means to initiate I/O and control the ILLIAC IV quadrant. Three modules of one operating system use HSS in the performance of their duties: the Execution Monitor, the Job Partner, and the Data Processor. HSS consists of a set of intrinsics imbedded into the B6500 MCP and some ALGOL code which must be compiled into the programs using HSS.

The following parts of HSS have been coded:

- Generation of ILLIAC IV I/O descriptors;
- Allocation of BIOM space;

- Modifications to the ALGOL compiler to allow invocation of the HSS intrinsics;
- A procedure to handle ILLIAC IV interrupts.

The following parts of HSS are being written:

- Error analysis including Transient Error Counters;
- Operator communication using the SPO (instead of DataCom as planned);
- List I/O using the list feature of the CDC (the list feature has not been implemented into hardware in the way it was described, thereby complicating its use);
- Generation of the scan out command to initiate the I/O.

2.1.2 Operating System II

Less effort was spent last quarter on Operating System II (OSII) than in the previous quarter (two key personnel were diverted to other activities); however, activity continued in:

- Further design and description of the extensions required in the FORTRAN language to support OSII;
- Study of the B6500 FORTRAN and ALGOL compilers and the B6500 MCP in preparation for making modifications;
- Study of means to integrate OSI and OSII into a composite whole.

2.2 Compilers and Translators

2.2.1 GLYPNIR

GLYPNIR now allows simple vectors and generalized pointer indexing. Work is progressing on I/O statements and better routing constructs. The writing of a GLYPNIR users manual and macro generator is progressing satisfactorily.

2.2.2 FORTRAN

In mid-February, a four man design team was formed to design the syntax and ILLIAC IV code specification for an extended version of FORTRAN. The design group is operating under two general premises in this design. The resulting language should be:

- as close to FORTRAN in syntax as possible, including standard FORTRAN IV (ASA) as a subset;
- simple enough to produce predictably efficient ILLIAC IV code.

Discussions were held with Burroughs about implementation of the design. Tentative arrangements for compiler implementation to be performed by Burroughs personnel at Paoli have been made.

The syntax design and detailed code specification are expected to be completed by the beginning of June. During the design phase users physically located at the University of Illinois will be consulted about the desirability of the syntax and proposed implementation. It is proposed that during June as many as possible of all potential users be given presentations on the language. The presentations will include hand generation of FORTRAN code for user selected sections of their proposed applications.

2.2.3 TRANQUIL

Evaluation and review of the TRANQUIL language and compiler continued during the first quarter of 1970. The cumulative evaluation of those within the project was presented to a group of nationally renowned software experts and future ILLIAC IV users on January 7th. The opinions of these experts were gathered both verbally at the review meeting in Chicago and from letters received subsequent to the meeting.

The opinions of these experts, together with an overall review of the project's aims and directions, led project management to conclude that the goals of the project would be better served by abandoning the TRANQUIL effort in favor of a fresh start on a FORTRAN language

especially extended for ILLIAC IV. Even though work on the TRANQUIL project ended on January 31, it is felt the experience gained by the personnel involved will prove invaluable on the FORTRAN effort.

2.3 Assembler

ASK was extended this quarter to include macros and conditional assembly features.

The macros have two noteworthy features:

1) They are text-substitution macros rather than macro instructions;
2) The programmer structures the calling line in the definition of the macro.

Example:

```
DEFINE  TO  &ADR; = -@@-1+@ADR;  ##;
SKIP,  TO  LOOPSTART;
```

The conditional features also deal with text and not statements, expressions, or any other syntactic entity.

Example:

```
&IF  P  &THEN  ADRN  &ELSE  SBRN  &FI  *X(2);
```

2.4 Utilities

Work is continuing on the peripheral conversion programs. The file security and machine use accounting systems are being studied and defined. ILLIAC IV FORTRAN intrinsic routines are being drafted and tested. Considerable thought has been given to the debugging facilities, although they have not been defined; work has commenced on an arithmetic scratch pad which will be part of the debugging system. Graphic display software is being discussed with the graphics and data communication group.

The instrumentation programs (to measure program performance) have been given a low priority due to the aforementioned projects.

2.5 Interactive Communications and Graphics

During the quarter, further investigations of commercial firms who supply both computer output microfilm recording systems and interactive graphics display terminal systems were conducted. On the basis of the investigations, formal bid specifications were written and the bids for the microfilm system were sent out. The bids for an interactive graphics terminal system will be sent out in the next quarter--in time to allow a final decision on both sets of equipment by the end of May.

Expected delivery is December 1, 1970 for microfilm and January 1, 1971 for the interactive graphics terminal. The basic qualifications of the microfilm recorder system are as follows:

- Very high quality precision Cathode Ray Tube (CRT) system displaying points, vectors, and characters. Not less than one thousand (1,000) lines/inch resolution on CRT device and 96 characters in character generator;
- Film transport system for both 16 and 35 mm films;
- Internal digital computer to allow highly flexible design of recording system software.

There are two versions of the interactive graphics display terminal in the bid proposals: 1) a minimal capability system and 2) an augmented capability system. The minimal version consists of:

- High quality CRT display system with points, vectors and characters;
- Internal digital computer to allow highly flexible design of graphics terminal software;
- Various manual input devices such as keyboard, lightpen, data tablet;
- Directly generated display images from core memory data list.

The augmented capability display terminal consists of the same requirements as the minimal display terminal system with the addition of:

- An interpretive display image technique by which a data structure contained in the terminal computer memory is interpreted during display,

thus allowing highly flexible arrangement between display data, the problem it is associated with, and necessary interactive capabilities required for the solution of the problem;

- Two-dimensional images derived from three-dimensional data. Either true perspective or orthogonal projection will be the method used (selected by the bidder).

In order to coordinate the development of graphics devices and utilities and interface them to the remainder of the system, a new service group was formed during the quarter encompassing all data communications and graphics utilities. The data communications group will develop the proposed data communications network, interface the ILLIAC IV system to the ARPA network, and develop and maintain the basic graphics utilities software packages to be developed.

2.6 B5500 Operation

	No. of Jobs	Process Hours
January	24,233	203.81
February	23,573	168.06
March	25,061	197.52

Machine use has been heavy and steady. The system has been operating three shifts a day, seven days a week.

In February, one processor and two tape units were removed. Diagnostics has been somewhat hampered by the removal of the processor; the Diagnostics Group has been using a great deal of machine time and will continue to use even more as ILLIAC IV nears completion.

Batch processing is receiving good turnaround time.

3. APPLICATIONS

3.1 Numerical Analysis

3.1.1 Numerical Solution of Problems in Hydrodynamics

The capabilities of the ILLIAC IV render it highly suitable for the solution of hydrodynamic problems in three dimensional space. Of particular interest is the study of flows for both intermediate and large Reynolds numbers. The co-ordinate system in which the problem is formulated is greatly influenced by the geometric and physical boundary conditions of the actual problem and by computational feasibility.

The set of quasi-linear partial differential equations governing the flow can be solved by finite difference techniques with appropriate boundary constraints depending on the nature of the physical problem and its particular geometry. Of singular importance in the solution of the flow problem governed by the partial differential equation system is the utilization of finite difference representations which are both stable and converge to the true solution of the physical problem. With the aim of solving the Navier-Stokes equations for the flow over cylindrical and spherical bodies, the following finite difference schemes were analyzed with regard to stability using the Von-Neumann analysis:

- Brailovskaya two step scheme;
- Cheng-Allen two step scheme;
- Dufort-Frankel and leap frog scheme.

These finite difference schemes, along with the Crank-Nicholson scheme, are undergoing numerical experimentation with Burger's equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}$$

Re = Reynolds number; u = flow velocity

to determine the most suitable scheme possessing the best properties of stability, rate of convergence, and tolerance to different initial

and boundary conditions. Modifications of these and other schemes will be considered. The optimum scheme is to be chosen for use with the full Navier-Stokes equations for the calculation of the flow over cylindrical and spherical bodies.

3.1.2 Eigenvalues

3.1.2.1 Matrix Storage Methods

3.1.2.1.1 Matrix Storage for Jacobi's Method

The modified Jacobi Method for finding eigenvalues of a symmetric matrix has been coded in assembly language, ASK, and is now being debugged.

One change was made in the program described last quarter [2]. Instead of making the matrix diagonally dominant, the N/2 superdiagonal elements, A[2I, 2I+1] I = 0, 1, 2, ..., N/2 - 1 (N order of matrix), are subjected to the test:

$$|A[2I, 2I + 1]| \leq 10^{-m};$$

where m is a power of two. This test is made at the beginning of each elimination process. If the test fails, elimination is completed, otherwise, skip to the 2nd row, 2nd column, shuffle, and test the A[2I, 2I+1] again. After N - 2 shuffles, the factor, m, for the above equation becomes 2 x m. The advantage of this test over the original test is that the largest A[2I, 2I+1] are eliminated first, thus achieving a faster convergence toward the eigenvalues.

For $m > 8$, the usual test for convergence (i.e., $\xi \leq 10^{-8} E_0^2/D_0^2$, where $\xi = E^2/D^2$ taken from the present matrix) is executed and, if satisfactory, convergence is reached.

3.1.2.1.2 Matrix Storage for QR-Algorithm

The modified QR method consists of two steps: 1) reduction of the original matrix to an upper Hessenberg form, and 2) the modified QR iterations which for an n x n matrix are as follows:

1) Computation of the first column of

$$\Gamma = (A_k - \xi_1^{(k)} I)\ (A_k - \xi_2^{(k)} I)\ \ldots\ (A_k - \xi_{n-1}^{(k)} I)$$

for n odd, and

$$\Gamma = (A_k - \xi_1^{(k)} I)\ (A_k - \xi_2^{(k)} I)\ \ldots\ (A_k - \xi_n^{(k)} I)$$

for n even. The matrices, A_i, are constructed by

$$A_{i+1} = Q_i^* A_i Q_i, \text{ where } A_i = Q_i R_i, \text{ i.e., } A_{i+1} = R_i Q_i$$

$$\text{for } i = 1, 2, \ldots$$

where A_i is the original matrix and R_i and Q_i are an upper triangular matrix respectively.

2) Finding of a unitary matrix N_1 such that the first column of $N_1^* \Gamma$ is e_1.

3) Reduction of $N_1^* A_k N_1$ to an upper Hessenberg form by unitary transformations according to whether N is odd or even. The original QR-algorithm calculates A_{k+2} from A_k using

$$\Gamma' = (A_k - \xi_k I)\ (A_k - \xi_{k+1} I)$$

in place of Γ in steps 1) and 2). This algorithm makes a matrix $N_1^* A_k N_1$ almost triangular except for three non-zero elements below the subdiagonal and the PE efficiency is fairly low at the computation in stage 3). In the modified QR, this situation is considerably improved. In addition, the modified algorithm reduces, in the course of iterations, the problem of finding eigenvalues of a matrix to that of finding eigenvalues of several matrices of smaller sizes which result from the partition of the original matrix. Thus, the algorithm saves a considerable amount of computation time. The sizes of submatrices and the speed of convergence largely depend upon the choice of ξ's, the origin shifts. The algorithm was tested with origin shifts chosen as:

1) eigenvalues of $\begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$

2) eigenvalues of $\begin{bmatrix} a_{i-1,i-1} & a_{i-1,i} \\ a_{i,i-1} & a_{i,i} \end{bmatrix}$

for $i = n, n-2, \ldots, n-2\rho$; $\rho = \left\lfloor \frac{n}{2} - 1 \right\rfloor$

where $\lfloor x \rfloor$ is the greatest integer which does not exceed x.

3) eigenvalues of $\begin{bmatrix} a_{(n-2q-1,n-2q-1)} & a_{n-2q-1,n-2q} \\ a_{n-2q,n-2q-1} & a_{n-2q,n-2q} \end{bmatrix}$

where $q = \left\lfloor \frac{n-3}{4} \right\rfloor$.

4) eigenvalues of $\begin{bmatrix} a_{i-1,i-1} & a_{i-1,i} \\ a_{i,i-1} & a_{i,i} \end{bmatrix}$

for $i = n, n-2, \ldots, n-2q$, where q is defined in 3).

In a limited number of experiments, the best result was obtained with the origin shifts of 4). Due to the importance of the choice of the origin shifts, further studies will be made.

A new storage scheme is developed for calculation of non-symmetric matrices for the modified QR method and Householder's method. If the order of the matrix is n, where $2^{i-1} < n \leq 2^i$ for some integer i, and m is the number of PE's in the machine, this scheme allows simultaneous access to each element of any $m/2^i$ consecutive rows or columns. The number of routings required for arranging elements into this scheme is found to be considerably smaller than that for $\sqrt{n}$ - skew storage. The mapping of an element a_{ij} into PE memory is

$$a_{ij} \rightarrow \text{loc } i,\ PE(2^k j_0 + 2^{k-1} j_1 + \ldots + j_k + i) \pmod{2^{k+1}}$$

where $j_0, j_1, \ldots, j_k$ are such that j is expressed by

$$j = 2^k j_k + 2^{k-1} j_{k-1} + \ldots + 2j_1 + j_0$$

and 2^{k+1} is the number of PE's in the machine.

3.2 Linear Programming

Design of the data preprocessor for the Linear Programming System (LPS) was largely specified this quarter. This preprocessor operates on input data already existing as B6500 files. Facilities for inputing the data (i.e., user specification and modification of LP matrices), will be separate from the LPS solution package and will interface with it. Following input, the major preprocessing tasks on the data are allocating rows to PE's, scaling the A-matrix, converting from B6500 to ILLIAC numeric representation, and formatting ILLIAC disk files. Due to the quantity of input data and the amount of manipulation required for the LP SETUP, most of this preprocessing will be done in the ILLIAC PE's rather than in the B6500. The B6500 allocates rows to PE's and passes to the ILLIAC disk an A-matrix file which is row, column, and PE sorted. The records of this file are in B6500 format and sized to interact well with the B6500 multiprocessing environment. The ILLIAC converts the data in this file to ILLIAC representation and re-formats the records to LPS size and specifications. Row and column scaling factors are calculated to equilibrate the A-matrix and to transform all upper bounds to a common value. Final output from this preprocessor is a set of scaled data files for the PRIMAL solution routine.

3.3 Long Codes

The programs mentioned last quarter are essentially complete. The primary program models the behavior of a discrete linear dynamic system of order ten or less. The user specifies the initial state of the system by a vector, $\vec{X}(0)$. The program then produces a discrete time series of new states, $\vec{X}(1)$, $\vec{X}(2)$, ... , $\vec{X}(TMAX)$, by the application of a series of transition matrices, $\Phi(1)$, $\Phi(2)$, ..., $\Phi(TMAX)$.

The transition matrices, Φ, are assumed to be non-derogatory matrices, that is, we assume that no two sub-blocks of the Jordan canonical form of Φ contain the same eigenvalue. This requirement must be met if Φ is to be transformed into the companion matrix form:

$$\left[\begin{array}{c|c} \begin{matrix} 0 & 0 & \cdots & 0 \end{matrix} & \Psi_1 \\ \hline I & \begin{matrix} \Psi_2 \\ \vdots \\ \Psi_n \end{matrix} \end{array}\right]$$

The search for a relation between this requirement and the commonly considered system properties of identifiability and observability was rewarded by the development of a proof which guarantees that the matrix, Φ, is non-derogatory if the system is identifiable, that is, if the matrix

$$B \text{ def } \left[\vec{X}(0) \mid \Phi \vec{X}(0) \mid \Phi^2 \vec{X}(0) \mid \cdots \mid \Phi^{n-1} \vec{X}(0)\right]$$

is nonsingular.

The user-specified transition matrix is used at each time step to simulate an invariant system. A set of transition matrices for a system with sinusoidally varying coefficients will be generated using a matrix power series expansion if the desired period is specified. At each time step, an observation vector, $\vec{Z}(K)$, is produced by applying a user-specified observation matrix to the corresponding state vector. The state vectors, the observation vectors, or both vectors may be contaminated by additive noise of specified variance if desired. The generated noise vectors have mutually independent components, each with essentially Gaussian distribution.

The time series of observation vectors provide test data for several newly developed algorithms. One subprogram estimates the period of the coefficients of a system with sinusoidally varying coefficients. It has been used successfully on outputs from systems contaminated by

noise vectors 1000 times the initial state vector magnitude. Two identification subprograms are also completed; an extension of the scalar identification algorithm of R. C. K. Lee to vector observations, and a stochastic approximation procedure that appears to satisfy the convergence criteria established in the literature of mathematical statistics, e.g., A. Dvoretzky [3], and J. H. Venter [4].

The new programs will be used to compare the effectiveness of the identification schemes under development with published results.

3.4 Large-Scale Planning

Work in this area is concentrated on the development of methodology for the design of generalized linear programming models, matrix-generators and report-generators, to facilitate the handling of the very large models expected to use the ILLIAC IV Linear Programming System.

General models can be designed for classes of resource-allocation problems. Such models, comprising the complete set of activities and constraints relevant for a given class of problems, are used as a framework within which models for specific problems are formulated.

Model formulation is by means of a matrix-generator program which accepts parameters and problem-specific data, and passes an internally-generated model to the ILLIAC IV linear programming code for solution.

Since normal linear programming output is generally intelligible only to the formulator of the problem, it is desired to interpret the solution before it becomes output. This is accomplished by a report-generator program which matches the internally-allocated row and column identifiers with user-assigned names, and then outputs the solution in a predefined format acceptable to decision-makers.

Although this effort is in its infancy, some progress has been made in the design of a general agricultural-sector model which will

serve as the test case. Some initial design of two matrix-generators, one model-specific and the other general, has been done. Future progress in these areas will be reported.

3.5 Meteorology

A topic has been selected for study that requires an upper limit on today's computing power. This topic is the study of the growth and decay of a thunderstorm cell. This will be a continuation in three dimensions of the recently completed one- and two-dimensional studies made at the Laboratory for Atmospheric Research at the University of Illinois. The project will involve consideration of the physical processes to be included in the model and the solution of the resulting system of equations by finite difference techniques for varying physical parameters. The three-dimensional study is appropriate for adaptation to ILLIAC IV (especially from a physical point of view since in a thunderstorm all the vertical and horizontal velocities are of equal magnitude).

3.6 Signal Processing

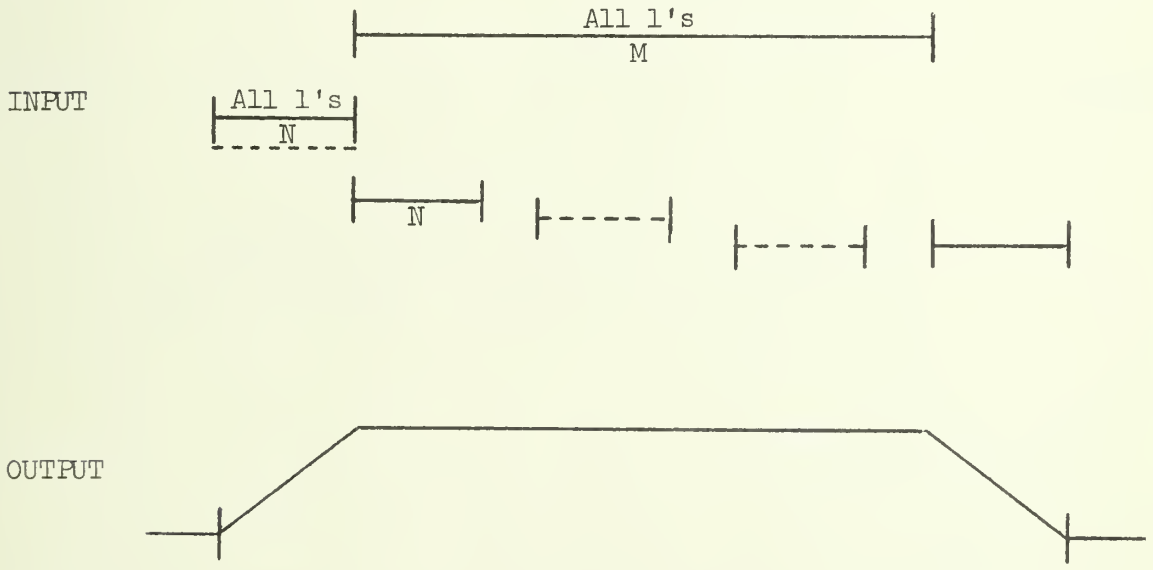
The Fast Fourier Transform subroutine has now been completely debugged. The subroutine can handle any number of data sets ranging in size from 8 to 4096 sample points. The restrictions are that all data must be core-contained (128 K maximum) and that the size of each data set must be an even power of two. A document explaining the program is now being generated and a short users manual will be produced. For the sake of efficiency, the program was written completely in assembly language using 32-bit mode floating point arithmetic. In general, each data point is represented as a 64-bit complex number, the outer portion being the real part and the inner portion being the imaginary part. Timing estimates are available and will be incorporated in the documentation.

Two of the basic programs for the Signal Processing System being developed for ILLIAC IV have been extended to give the user flexibility in selecting the type of output desired and the mode of computation required.

There are two techniques available to the user; the technique selected is dependent upon the length of the input vectors. The first algorithm is to be used for those cases in which the input data vector for an auto-correlation (or data vectors for a cross correlation) does not exceed 1800 words. This program stores each autocorrelation vector or both cross-correlation vectors in a single processing element and can perform 64 auto or cross-correlations simultaneously in 64-bit mode. It can perform 128 auto or cross correlations simultaneously in 32-bit mode by inputing multiplexed data. The second technique stores the input data across the processing elements in rows of 64 data words. A data string of 3000 values requires 47 rows of memory for storage. Since each word of memory is 64-bits in size, two input vectors can be multiplexed in 32-bit mode. The result is the calculation of two functions at the same time.

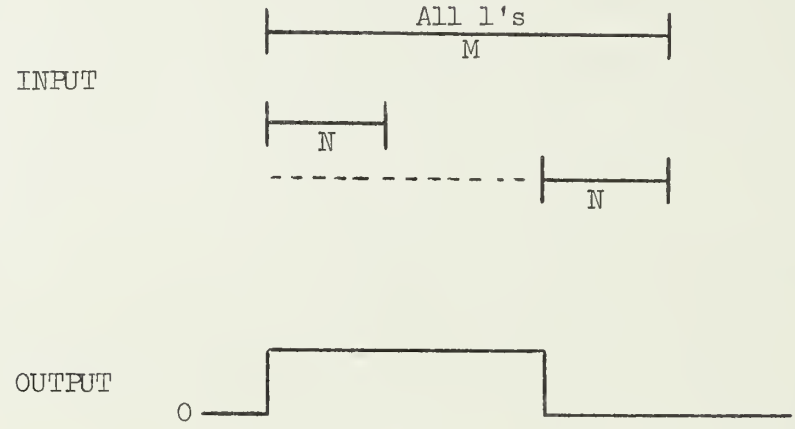
The program has been extended to give the user the capability of selecting output vector lengths using the following options:

1) Transient unnormalized with taper at either or both ends:



The length of the output vector with taper at both leading and trailing edge is M + N - 1.

2) Transient normalized with no taper at either end:



The length of the output vector is M - N + 1.

The user may designate the desired operation to be performed:

- Autocorrelation
- Cross-correlation
- Convolution

and may specify the starting point, ending point, and number of lags.

Programs currently under development are:

- Normal moveout corrections--used in correcting non-vertical ray paths to vertical ray paths;
- Stacking--used to sum several records and create a single record.

3.7 ILLIAC IV Education

3.7.1 FOURUM

The ILLIAC IV Users Group, FOURUM, will become the main vehicle for disseminating information to potential ILLIAC users. The mailing

list for the FOURUM Newsletter has been extended to include all major departments within the University of Illinois so that the total University community can be aware of the potentialities of ILLIAC.

Additionally, new documents and new FOURUM Communications will be announced in the Newsletter--the user interest categories will be discontinued. Thus, FOURUM members are not constrained to fixed interest categories--they may order the documents or communications they wish to receive.

3.7.2 Documentation

Work continues on the automation of the FOURUM mailing list which will be complete in May 1970.

4.1 Administration and Services

Budgeted expenditures through the third quarter of fiscal year 1970 are as follows:

Burroughs Corporation	$21.5 million
University of Illinois	$ 4.7 million

The fiscal figures for both the University of Illinois and Burroughs Corporation for March will be available near the end of April. Therefore the following Project expenditures and commitments incurred through February, 1970 are presented:

Burroughs Corporation (Expenditures and Commitments)	$21.8 million
University of Illinois (Expenditures and Commitments)	$ 4.4 million

An analysis of the required funds for the first half of fiscal year 1971 (July-December) is under way; the results will be published in the next Quarterly Progress Report.

The ILLIAC IV Project has a personnel strength of 127 people:

Professionals	35
Nonacademic	21
Research Assistants	34
Hourlies	30
Illini Girls (Secretarial Assistants)	7

with gradual increases planned through a six month period following delivery of the system.

Separate funding studies are currently under way for the Laser Mass Storage System.

The Center for Advanced Computation building for the project is in skeletal form. It has a target completion date of October, 1970.

At this time, the local craftsmen are on strike; however, it is anticipated that there will be a settlement soon. No serious delay affecting the target date is forecasted.

The University of Illinois Faculty Senate has approved the establishment of the Center for Advanced Computation as the academic unit operating the ILLIAC IV system. Subject to final approval from the University of Illinois Board of Trustees and the State of Illinois Board of Higher Education, the Center for Advanced Computation should be formally established in mid-June, 1970.

REFERENCES

[1] A. Hashimoto, L. Abel, and C. Tanaka, "Improvement of Section Simulator Efficiency by Partitioning Board Simulators," ILLIAC IV Document No. 216, (to appear).

[2] ILLIAC IV Quarterly Progress Report, October, November, and December, 1969. ILLIAC IV Document No. 204, DCS Report No. 383. Urbana, Illinois: Department of Computer Science, University of Illinois, 1970.

[3] A. Dvoretzky, "On Stochastic Approximation," Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 39-55, 1956.

[4] J. H. Venter, "On Dvoretzky Approximation Theorems," Annals of Mathematical Statistics, Vol. 37, pp. 1534-1544, 1966.

DOCUMENTS

Marks, R. E., "On the Machine Conversion of B5500 Algol to CDC 6000 Algol," ILLIAC IV Document No. 207, DCS File No. 821, (January 30, 1970).

Yamamoto, T., and Lermit, J., "Gradient Projection Method for Linear Programming," ILLIAC IV Document No. 205, DCS File No. 820, (January 21, 1970).

Parker, James L., "Data Management on ILLIAC IV," ILLIAC IV Document No. 206, DCS File No. 822, (January 21, 1970).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
ILLIAC IV QUARTERLY PROGRESS REPORT
January, February, March 1970

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
January-March, 1970 - Progress Report of the ILLIAC IV Project

5. AUTHOR(S) *(First name, middle initial, last name)*

6. REPORT DATE
April 15, 1970

7a. TOTAL NO. OF PAGES
33

7b. NO. OF REFS
4

8a. CONTRACT OR GRANT NO.
USAF 30(602)-4144

b. PROJECT NO.
46-26-15-305

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
ILLIAC IV Document No. 208
DCS Report No. 395

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
RADC TR

10. DISTRIBUTION STATEMENT
Qualified requesters may obtain copies of this report from DCS.

11. SUPPLEMENTARY NOTES
NONE

12. SPONSORING MILITARY ACTIVITY
Rome Air Development Center
Griffiss Air Force Base
Rome, New York 13440

13. ABSTRACT

See the Report Summary on Page 1 within the Report itself.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Design and Construction						
Hardware Diagnostics						
Components and Circuits						
Supervisory Systems						
Compilers and Generators						
ASK						
GLYPNIR						
TRANQUIL						
Utility Programs						
Interactive Communications and Graphics						
Storage Allocation						
Numerical Analysis						
Ordinary and Partial Differential Equations						
Matrix Algebra						
Linear Programming						
Combinatorial and Discrete Mathematics						
Meteorology						
Earth Sciences						
Signal Processing						